Technical white paper

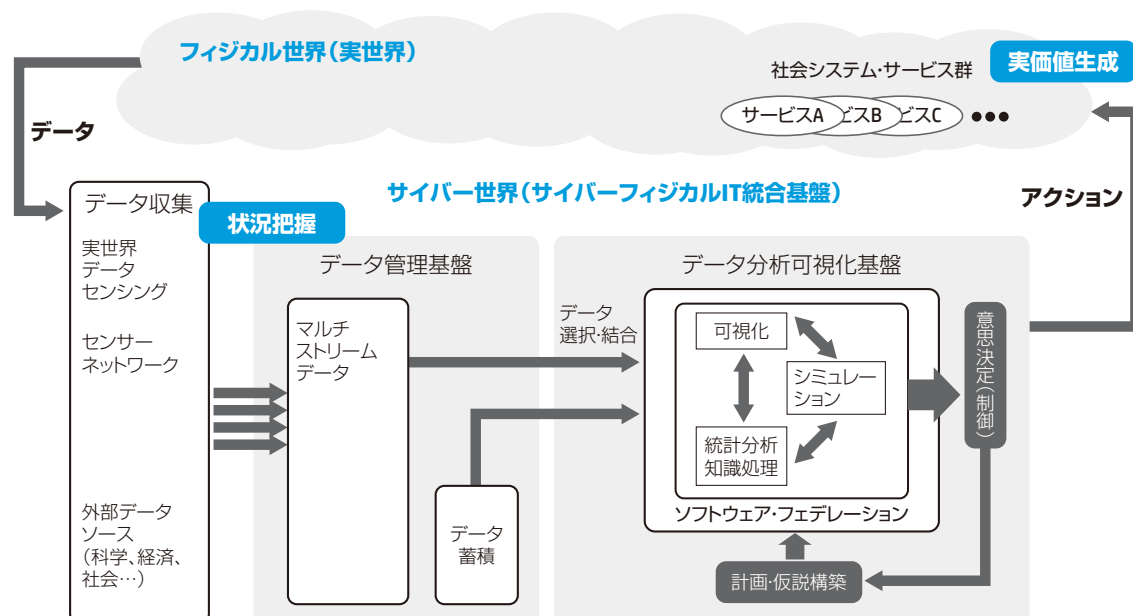
No SQLデータベース MongoDBのスケーラビリティ検証

北海道大学知識メディアラボラトリーでは、膨大な数のセンサーから集まってくる多種多様なデータで構成されるフィジカル世界からの情報を、計算機上のサイバー世界で分析・処理し、再び人間の暮らすフィジカル世界に役立てることを目指すCPS(Cyber Physical System)の研究に取り組んでいる。この研究では、「ビッグデータ」とも呼べる巨大な量のソースデータから、人間が評価や判断を下すための材料をいかに抽出・提示できるかが、技術的に重要な鍵を握っており、こうした処理の高速化を実現するために大規模な並列処理のIT環境を構築しようとしている。このホワイトペーパーでは、同ラボラトリーの特任助教、猪村 元氏が行った、HP ProLiant DL980 G7、およびNo SQL系データベースとして注目を集めるMongoDBを組み合わせたシステムによる並列処理パフォーマンスのスケーラビリティ検証を紹介。このシステムがビッグデータの処理基盤として、大きな可能性を持っていることが証明された。

1. CPSで扱うデータは極めて膨大で多種多様

現代社会では、日常の様々な場所、あるいは機器に極めて大量のセンサーが設置されており、そこで刻々と収集される情報はITシステムというサイバーな世界に集約され、現実(フィジカル)世界での出来事の一面を把握するために活用されている。北海道大学知識メディアラボラトリーは、こうした状況をさらに発展させ、センサーとITシステムなどで構成されるサイバーな世界と人間の暮らすフィジカルな世界を融合して、エネルギーや環境、防災対策などの社会システムの高度化や最適化、効率化に役立てようという、CPS(Cyber Physical System)の研究に取り組んでいる。

CPSでは、センサーが捉えた極めて大量の、しかも多種多様なソースデータを、複数の個別システムをまたいでリアルタイムに収集し、統一的に集約・分析。その結果を評価したうえで、情報発信などの形で現実世界へフィードバック。現実世界の変化を再びセンサーでキャッチしてサイバー世界へ送り込む、というサイクルを繰り返す(図1)。集約・分析された情報の評価作業は人間の手によって行われるが、収集されるソースデータの量はこれまでの常識をはるかに超える。その規模は、従来、データマイニングなどで扱っていた量と比べ2桁、100倍ほど大きくなる。まさに、「ビッグデータ」を扱うことになるのである。

図1. CPSのアーキテクチャー

ビッグデータから迅速に的確な評価を下せるようにするためには、データの集約・分析結果を人間が直感的に把握できるようにする「データ分析可視化基盤」の実現が鍵となる。知識メディアラボラトリーでは、このデータ分析可視化基盤の技術を確立するために、研究インフラとして高い処理性能と優れた信頼性、拡張性を備えたインテル® Xeon® プロセッサー E7ファミリーを8基(80コア)搭載したマルチプロセッサーサーバー、HP ProLiant DL980 G7を中心としたシステムを構築している。今回の検証でもこのシステムを活用して、オープンソースのデータベースアプリケーション、MongoDBでのパフォーマンスのベンチマークテストを行った。

2. 低コストで極めて高い柔軟性を実現できるMongoDB

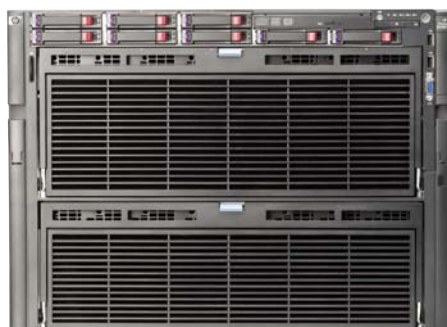
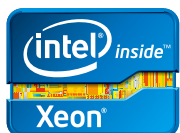
同ラボラトリーが目標とする具体的な成果は、札幌市の除排雪システムの最適化やドライバーの適切な誘導などを実現するためのデータ分析可視化基盤を開発することだ。その最初のステップとして、ソースデータを基に、人間が評価するための可視化された情報を作成するまでの処理フローを検討している。ソースデータとなるのは、まず各種センサーを搭載して札幌市内を走行するプローブ・カーから刻々と送られてくる日時や位置、走行スピード、進行方向などから成るストリームデータ、これに加え、過去に蓄積してきた走行データも使用する。これらのソースデータをHP ProLiant DL980 G7上で稼働するデータベースに取り込み、メッシュなどの単位で集約演算・平滑化などの計算処理を並列で実施。この処理結果から可視化を行う、というプロセスを考えている。
ベンチマークテストでは、プローブ・カーからのデータと過去の蓄積データを用い、8基80コアを搭載したHP ProLiant DL980 G7上のMongoDBで行う計算処理を並列化、コア数を増やしていくことで処理時間がどのように変化するかを計測した。用意したソースデータの規模は約270GB。エントリー数でいうと約5億3000万になる。
このソースデータはHP ProLiant DL980 G7に内蔵でき、892,000 IOPS(Read)の超高速なI/Oパフォーマンスを実現するHP PCIe IOアクセラレータ G2 for ProLiantサーバー(以下、IOアクセラレータ G2)上に展開。MongoDBで並列計算処理を行わせた(図2)。データベースアプリケーションとしてMongoDBを選択したのは、以下のような理由による。

1)システムコストを低く抑えられること
MongoDBはオープンソースで開発されており、処理のためのプログラムさえ自分たちで用意できれば、一般的な商用RDBMSと比べて、データベースシステム構築のためのコストを大幅に削減することが可能になる。ただし、NUMAには非対応である。

2)No SQLであり、スキーマ定義が不要なこと
MongoDBは、データベースアプリケーションとして現在普及しているRDBMSと異なる、No SQL(Not Only SQL)に分類されるドキュメント指向型のデータベースである。No SQLは、RDBMSで必要なスキーマ定義をしなくて済むことが大きな特長の一つだ。今回のベンチマークテストでは、主にプローブ・カーからの走行データを用いている。しかし、今後、実用度を高めていくために除排雪車の稼働情報、バスなどの公共交通機関の走行情報、そして人間の歩行情報などもソースデータとして取り込んでいく予定である。また、ソースデータの属性の中でどこに注目すべきか、といったこともトライアル&エラーを繰り返しながら探っていこうとしている。このように、多種多様かつ膨大なデータ、どのようなデータ構造で取得できるかの予測もつきにくいデータなどを一括して扱える点で、スキーマ定義に縛られない高い柔軟性を備えたMongoDBには大きなアドバンテージがある。

3)インサートとリードで高いパフォーマンスを発揮すること
一般的なデータベースの場合、すでにデータベース内に保存されているデータのアップデートを頻繁に繰り返すという操作を行うケースが多い。このため、一般の商用RDBMSはアップデート操作で高いパフォーマンスを発揮できるようにデザインされている。これに対し、MongoDBはインサートとリードを高速処理できる設計になっている。開発を進めているデータ分析可視化基盤のデータベースは、インサートとリードの操作が大きなウェートを占めるという処理特性を持っている。こうした面からもMongoDBは研究テーマとマッチする。

図2. 検証で使用したシステム環境

インテル® Xeon® プロセッサー E7 ファミリー

ハードウェア:**HP ProLiant DL980 G7**
搭載プロセッサー:インテル® Xeon® プロセッサー E7-4870 2.4GHz 8基/80コア
搭載メモリー:2TB
搭載記憶領域:900GB HDD×8台　合計7.2TB
オプション:HP PCIe IOアクセラレータG2 for ProLiantサーバー 365GB×1台
OS:Red Hat Enterprise Linux 6.3
データベース:MongoDB 2.4.4

3. 標準的な処理プロセスでは40コア以上でパフォーマンスが低下

前述したシステム環境で、約270GBのソースデータを用い、計算処理に割り当てるコア数を増やしていったときの計算時間の推移を計測した結果が、図3のグラフである。10回処理を行い、その平均値を計算時間として採用した。MongoDBのエンジンに10コアが占有されるため、理想的には最大70コアまでパフォーマンスがスケールすることが期待された。
しかし、グラフから分かるように、計算時間は30コアで最短となっている。それ以上コア数を増やしてもさらなる短縮は図れず、逆に計算時間は延びてしまう、パフォーマンスは低下する、という結果となった。こうした現象の原因を探った結果、40コア近辺からカーネル処理の負荷が増大。カーネル処理がボトルネックとなり、カーネル時間が増加してしまうことが分かった（図4）。
このため、カーネル時間の増大を回避するための工夫を施し、改めてベンチマークを取ることにした。

図3. 標準的な検証環境での計算時間の推移

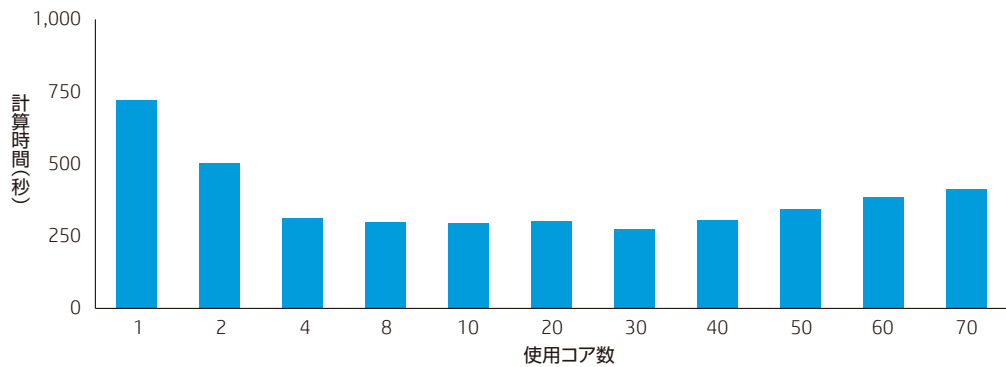
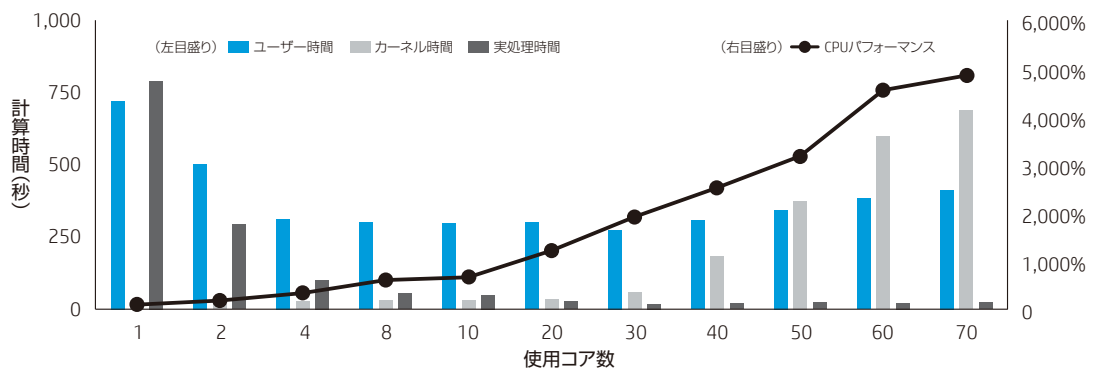



図4. スケールしない原因はカーネル時間の増加

4. チューニングを施すことでスケーラビリティが大幅に向上

カーネル時間の増大を回避するための工夫として実施したチューニングは、以下のように大きく二つある。

1）スーパバイザ・プロセスを二つ用意

スーパバイザ・プロセスの数を一つで実行させると、NUMA対応していないMongoDBは40コアから（ソケット数が4を超えてしまうと）カーネル処理時間が長くなってしまう。この問題を回避するため、スーパバイザ・プロセスの数を二つにし、それぞれを40コア（4ソケット）に割り当てることでカーネル処理の効率を向上させることができ、結果としてMongoDBの並列処理性能を向上させることができた。実際は各スーパバイザ・プロセスには35コアずつを割り当て、10コアはMongoDBのエンジンに割り当てとなる。スーパバイザ・プロセスのプログラムはJavaを使って作成した。

2）I/0プロセスをプロデューサー&コンシューマパターンでチューニング

検証を行った処理を詳細に検討すると、プロセッサーが実行する計算プロセスの負荷が重く、I/0プロセスにかかる負荷は軽いと想定された。このため、せっかく迅速にソースデータを読み出しても、計算プロセスが完了するまでI/0プロセスは待機状態を余儀なくされる。そこで二つのプロセスを切り離し、読み出しを非同期で実施、計算プロセスにより多くのリソースを割り当てられるようにスケジューリングの調整を行った。その際、I/0プロセスの制御には、並列系プロセスの制御法として良く知られる「プロデューサー&コンシューマパターン」を採用。I/0を行うプロセスと計算を行うプロセスの数のバランスを取ることで処理の効率化を図った。
こうした二つのチューニングを施して計測した計算時間の推移を示したのが図5である。コア数を増やすにつれて計算時間は順調に短くなっており、処理に割り当て可能な最大値の70コアまで、プロセッサーリソースを無駄なく使い切れていることが分かる。1コアでの計算時間を1として、コア数を増やしたときの処理効率を示した図6のグラフからも分かるように、70コアでは40

倍を超える驚異的なパフォーマンスを確認できた。
NUMAに非対応のMongoDBを使った並列処理で、このように想定以上のスケーラビリティを実現できたのは、実施した二つのチューニングに加え、超高速I/Oが可能なIOアクセラレータ G2を使ったことも大きく貢献していると、猪村特任助教は考えている。

図5. チューニング実施後の計算時間の推移

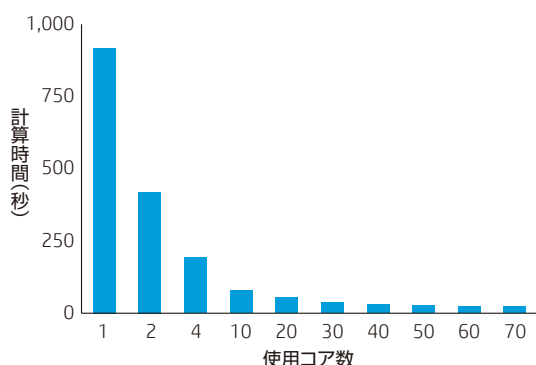
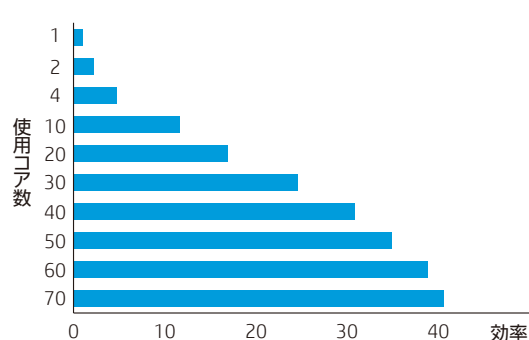



図6. チューニング実施後はコア数に比例して処理効率がスケール

5. HP ProLiant DL980 G7とMongoDBで、コスト効率良くビッグデータの処理基盤構築が可能

今回の検証では、HP ProLiant DL980 G7とIOアクセラレータ G2、MongoDBを使い、スーパバイザ·プロセスを二つにする、I/Oプロセスを調整し負荷の重い計算プロセスへリソースをより多く割り当てる、といったチューニングを施すことにより、70コアまでデータベースの並列処理パフォーマンスが確実にスケールすることを確かめることができた。
従来であれば、この規模の並列処理を実現しようとすると、クラスタ構成を選ぶしか道がなかった。しかし、この構成は運用管理の負担が非常に大きく、スケールアウトの規模拡大やOSなどのアップデート対応に際してプログラムの書き換えなども発生する。このため専任のエンジニアがいないと、リソースの利用率を高めたり、安定した稼働環境を維持したりすることは難しい。
「今回のようなスケールアップの環境であればシステム構成は非常にシンプルであり、研究の合間を使って運用管理の作業を行うことが可能です」と猪村特任助教は語る。「オープンソースのMongoDBを使うことでコストも抑えられます。大規模なシミュレーションや機械学習をはじめ、ビッグデータを扱う際の基盤としてHP ProLiant DL980 G7とMongoDBの組み合わせは非常に有効だと思います」（猪村特任助教）
もちろん、今回の検証成果は知識メディアラボラトリーが進めるデータ分析可視化基盤の開発でも大きな意味を持つ。「1コアでは15分ほどかかった処理が、70コアで並列処理させると、わずか20秒程度に大幅に時間を短縮できます。まだ試行錯誤の段階であるデータ分析可視化基盤の開発では、思いついたアイディアをどれだけ数多く試せるかが非常に重要です。また、パフォーマンスが高ければ、より長期間のソースデータを使って解析が行えるようになり、分析の精度が上がることにもつながります。今回の検証結果を心強く思っています」と猪村特任助教の期待は大きい。

安全に関するご注意　ご使用の際は、商品に添付の取扱説明書をよくお読みの上、正しくお使いください。水、湿気、油煙等の多い場所に設置しないでください。火災、故障、感電などの原因となることがあります。

お問い合わせはカスタマー・インフォメーションセンターへ
03-5749-8328 月～金 9:00～19:00　土 10:00～17:00（日、祝祭日、年末年始および5/1を除く）
機器のお見積もりについては、代理店、または弊社営業にご相談ください。
HP ProLiantに関する情報は http://www.hp.com/jp/proliant

〒136-8711 東京都江東区大島2-2-1